

Submitted in 09/022,979

DIALOG(R)File 351:Derwent WPI  
(c) 2003 Thomson Derwent. All rts. reserv.

009999442      \*\*Image available\*\*

WPI Acc No: 1994-267153/ 199433

XRPX Acc No: N94-210372

**Parameter setting method for serial port interface for printer - uses  
parameter table printer to set up data communication parameter using host  
computer**

Patent Assignee: OKI ELECTRIC IND CO LTD (OKID )

Number of Countries: 001 Number of Patents: 001

Patent Family:

| Patent No  | Kind | Date     | Applicat No | Kind | Date     | Week     |
|------------|------|----------|-------------|------|----------|----------|
| JP 6195280 | A    | 19940715 | JP 92356717 | A    | 19921222 | 199433 B |

Priority Applications (No Type Date): JP 92356717 A 19921222

Patent Details:

| Patent No  | Kind | Lan Pg | Main IPC      | Filing Notes |
|------------|------|--------|---------------|--------------|
| JP 6195280 | A    |        | 5 G06F-013/00 |              |

Abstract (Basic): JP 6195280 A

The setting method sets system parameter of the RS232 serial port interface using the system parameters setting processor (14). The parameter table from RS232 is transmitted to the host computer (11) from the printer.

The host computer, sets up a data communication parameter using data communication parameter determination processor (15) and the received parameter table. The data communication parameter is then transmitted to the printer. Based on this received data communication parameter, the parameter of RS232 is set up automatically between the host computer and printer.

ADVANTAGE - Makes communication possible between computer and peripheral devices. Eliminates need for user to set up parameter. Simplifies parameter setting process. Sets up parameter using minimum bits for transmission. Shortens real communication time. Facilitates data transmission and reception between arbitrarily chosen host computers.

**This Page Blank (uspto)**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-195280

(43) 公開日 平成6年(1994)7月15日

|                           |         |         |               |         |
|---------------------------|---------|---------|---------------|---------|
| (51) Int.Cl. <sup>5</sup> | 識別記号    | 庁内整理番号  | F I           | 技術表示箇所  |
| G 0 6 F 13/00             | 3 5 3 V | 7368-5B |               |         |
| H 0 4 L 29/10             |         | 7240-5K | H 0 4 L 13/00 | 3 0 9 A |

審査請求 未請求 請求項の数 3 (全 16 頁)

(21) 出願番号 特願平4-356717

(22) 出願日 平成4年(1992)12月22日

(71) 出願人 000000295

沖電気工業株式会社  
東京都港区虎ノ門1丁目7番12号

(72) 発明者 山口 辰己

東京都港区虎ノ門1丁目7番12号 沖電気  
工業株式会社内

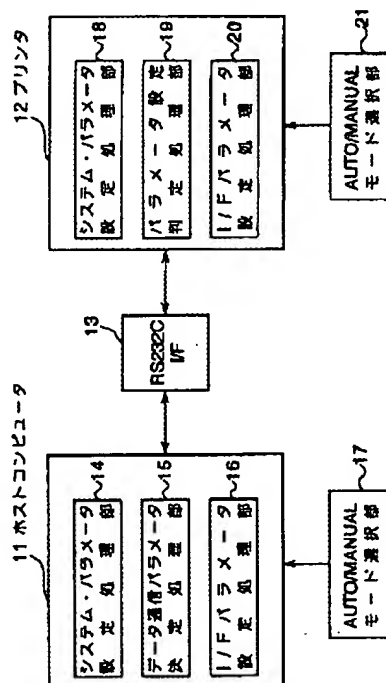
(74) 代理人 弁理士 船橋 国則

(54) 【発明の名称】 シリアルインタフェースのパラメータ設定方法

(57) 【要約】

【目的】 RS232C I/Fのパラメータ設定を簡便化し、初心者ユーザにも簡単に使用可能としたシリアルインタフェースのパラメータ設定方法を提供する。

【構成】 システム・パラメータ設定処理部14によってRS232C I/F13のパラメータを予め決められたシステム・パラメータに設定して通信可能な状態にし、この状態においてプリンタ12からホストコンピュータ11へRS232C I/F13のパラメータテーブルを送信し、ホストコンピュータ11では受信したパラメータテーブルに基づいてデータ通信パラメータ決定処理部15にてデータ通信パラメータを決定してこのデータ通信パラメータをプリンタ12へ送信し、このデータ通信パラメータに基づいてホストコンピュータ11とプリンタ12の間で自動的にRS232C I/F13のパラメータを設定する。



第1実施例を示す機能ブロック図

## 【特許請求の範囲】

【請求項1】 ホストコンピュータとターゲットマシンとの接続に使用されるシリアルインタフェースのパラメータ設定方法であって、

前記シリアルインタフェースのパラメータを予め決められたシステム・パラメータに設定して通信可能な状態にし、

この通信可能な状態において前記ターゲットマシンから前記ホストコンピュータへ前記シリアルインタフェースのパラメータテーブルを送信し、

前記ホストコンピュータでは受信した前記パラメータテーブルに基づいてデータ通信パラメータを決定してこのデータ通信パラメータを前記ターゲットマシンへ送信し、

前記ターゲットマシンでは受信した前記データ通信パラメータが設定可能であればその旨を前記ホストコンピュータに返答するとともに、前記データ通信パラメータにしたがって前記シリアルインタフェースのパラメータを設定し、

前記ホストコンピュータでは前記返答を受信した場合に前記データ通信パラメータにしたがって前記シリアルインタフェースのパラメータを設定することを特徴とするシリアルインタフェースのパラメータ設定方法。

【請求項2】 請求項1記載のシリアルインタフェースのパラメータ設定方法において、

アプリケーションが出力するデータの内容を解析し、

その解析結果に基づいて前記アプリケーションからの出力データ毎に前記シリアルインタフェースのパラメータを変更することを特徴とするシリアルインタフェースのパラメータ設定方法。

【請求項3】 ホストコンピュータとプリンタとの接続に使用されるシリアルインタフェースのパラメータ設定方法であって、

前記プリンタでは、前記ホストコンピュータが出力した印刷データから前記ホストコンピュータ側のシリアルインタフェースのパラメータを解析し、この解析結果に合わせてパラメータを設定することを特徴とするシリアルインタフェースのパラメータ設定方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、シリアルインタフェース(I/F)のパラメータ設定方法に関し、特にホストコンピュータとプリンタ等のターゲットマシンとの接続に使用されるシリアルI/Fのパラメータを設定する方法に関するものである。

## 【0002】

【従来の技術】 ホストコンピュータとその周辺機器である例えばプリンタとを接続するシリアルI/F(例えば、RS232C I/F)のパラメータを設定する方法として、従来、以下に説明する手法が採られていた。図17は、

ホストコンピュータ上で動作するプログラムの従来例を示すフローチャートである。まず、ホストコンピュータの起動と同時に、DOS(Disk Operating System)によってホストコンピュータを初期化し(ステップS201)、続いてSPEEDコマンドによってRS232C I/Fドライバを起動することにより、RS232C I/F回路およびRS232C I/Fプログラムの初期化を行ってRS232C I/Fを使用可能にする(ステップS202)。

【0003】 次に、ボー・レート(BAUD RATE)の選択を行う(ステップS203)。この処理ステップでは、75/150/300/600/1200/4800/9600のいずれかのボー・レートを選択する。次に、データ・ビット(DATA BIT)の選択を行う(ステップS204)。この処理ステップでは、8 bits/7 bitsのいずれかのデータ・ビットを選択する。次に、パリティ(PARITY)の選択を行う(ステップS205)。この処理ステップでは、NONE/EVEN/ODDのいずれかのパリティを選択する。

【0004】 次に、ストップ・ビット(STOP BIT)の選択を行う(ステップS206)。この処理ステップでは、1 bit/2 bitsのいずれかのストップ・ビットを選択する。次に、フロー・コントロール(FLOW CONTROL)の選択を行う(ステップS207)。この処理ステップでは、DTR HI/XON XOFFのいずれかのフロー・コントロールを選択する。以上の一連の処理により、ホストコンピュータ側では、RS232C I/Fによる送受信が可能となる。

【0005】 図18は、プリンタにおける制御部のブロック図である。図18において、31はCPU、32はホストI/Fなどの設定を行う場合に使用される液晶表示部(LCD)、33は操作部、34はRAM、35はプログラムROM、36はプリンタ部、37はRS232Cドライバ、38はRS422ドライバである。インタフェースであるRS232Cドライバ37およびRS422ドライバ38は、それぞれRS232Cコネクタ37aおよびRS422コネクタ38aを有している。

【0006】 図19は、プリンタ部36における操作部の正面図である。図19において、39はプリンタをオンライン状態もしくはオフライン状態にするためのON-LINEキー、40はオフライン状態のときメニューモードに入るためのMENUキー、41は選択された設定値を決定するためのENTERキー、42はメニューモードにおいて、選択可能な項目を1つ後ろへ戻すための「-」キー、43はメニューモードにおいて、選択可能な項目を1つ前へ進めるための「+」キーである。

【0007】 次に、プリンタ側での従来の処理動作について、図20および図21のフローチャートにしたがって説明する。まず、現在選択されているI/Fに受信データがあるか否かを判断し(ステップS211)、受信データがある場合は印刷処理を行う(ステップS212)。受信データがない場合は、ON-LINEキー39が押されたか否かを判断する(ステップS213)。ON-LI

NEキー39が押されていないと判定した場合は、OFF LINE表示を行い(ステップS214)、続いてMENUキー40が押されたか否かを判断する(ステップS215)。

【0008】MENUキー40が押されていないければ、再びON-LINEキー39が押されたか否かを判断し(ステップS216)、ON-LINEキー39が押されていないと判定した場合は、ステップS215に戻る。ステップS213およびステップS216でON-LINEキー39が押されたと判定した場合は、ON-LINE表示を行う(ステップS217)。ステップS215でMENUキー40が押されたと判定した場合は、MENUモードの表示を行い(ステップS218)、続いてホストI/FとしてRS232C I/Fが選択されたか否かを判断する(ステップS219)。この選択は、Parallel I/FもしくはRS232C I/Fのどちらかを「-」キー42又は「+」キー43で選択することによって行われる。

【0009】ステップS219でRS232C I/Fの選択でないと判定した場合には、Parallel I/Fの選択であることから、Parallel I/Fを設定する(ステップS220)。一方、RS232C I/Fの選択であれば、まず、ボー・レート(BAUD RATE)の選択を行う(ステップS221)。この処理ステップでは、75/150/300/600/1200/4800/9600のいずれかのボー・レートを選択する。次に、フロー・コントロール(FLOW CONTROL)の選択を行う(ステップS222)。この処理ステップでは、DTR HI/XON XOFFのいずれかのフロー・コントロールを選択する。

【0010】次に、データ・ビット(DATA BIT)の選択を行う(ステップS223)。この処理ステップでは、8 bits/7 bitsのいずれかのデータ・ビットを選択する。次に、パリティ(PARITY)の選択を行う(ステップS224)。この処理ステップでは、NONE/EVEN/ODDのいずれかのパリティを選択する。次に、ストップ・ビット(STOP BIT)の選択を行う(ステップS225)。この処理ステップでは、1 bit/2 bitsのいずれかのストップ・ビットを選択する。

【0011】以上の一連の処理により、プリンタがRS232C I/Fからデータを受信できるようになる。このようにして、ホストコンピュータとこれにRS232C I/Fを介して接続されるターゲットマシンであるプリンタとを、同じパラメータ(設定値)に設定することにより、ホストコンピュータからデータを受信し、プリンタでその受信データを印刷することが可能となる。

【0012】

【発明が解決しようとする課題】しかしながら、上記従来のパラメータ設定方法では、設定するパラメータが数多く存在する上に、データを正確に送受信するためにはホストコンピュータ側とプリンタ側の双方のパラメータを全く同じように設定しなければならないため、ユーザがパラメータを設定する際に間違いが生じやすく、特に、初心者ユーザにあっては、設定の複雑さからRS232C

I/Fの使用を敬遠しがちであった。本発明は、上記課題に鑑みてなされたものであって、RS232C I/Fのパラメータ設定を簡便化し、初心者ユーザにも簡単に使用できるようにしたシリアルインタフェースのパラメータ設定方法を提供することを目的とする。

【0013】

【課題を解決するための手段】本発明は、ホストコンピュータとターゲットマシンとの接続に使用されるシリアルインタフェースのパラメータ設定において、シリアルインタフェースのパラメータを予め決められたシステム・パラメータに設定して通信可能な状態にし、この通信可能な状態においてターゲットマシンからホストコンピュータへシリアルインタフェースのパラメータテーブルを送信し、ホストコンピュータでは受信したパラメータテーブルに基づいてデータ通信パラメータを決定してこのデータ通信パラメータをターゲットマシンへ送信し、ターゲットマシンでは受信したデータ通信パラメータが設定可能であればその旨をホストコンピュータに返答するとともに、データ通信パラメータにしたがってシリアルインタフェースのパラメータを設定し、ホストコンピュータでは上記返答を受信した場合にデータ通信パラメータにしたがってシリアルインタフェースのパラメータを設定する。

【0014】また、上記のパラメータ設定方法において、アプリケーションが出力するデータの内容を解析し、その解析結果に基づいてアプリケーションからの出力データ毎にシリアルインタフェースのパラメータを変更する。さらに、ターゲットマシンとしてプリンタを用いたシステムにおいて、プリンタでは、ホストコンピュータが出力した印刷データからホストコンピュータ側のシリアルインタフェースのパラメータを解析し、この解析結果に合わせてパラメータを設定する。

【0015】

【作用】シリアルインタフェースのパラメータを予め決められたシステム・パラメータに設定して通信できるようにし、この通信可能な状態において、ターゲットマシンから受信したパラメータテーブルに基づいてデータ通信パラメータを決定してこれをターゲットマシンに送信し、このデータ通信パラメータに基づいてホストコンピュータとターゲットマシンとの間で自動的にシリアルインタフェースのパラメータを設定する。これによれば、ユーザがそれぞれの機器のシリアルインタフェースのパラメータを設定しなくても通信が可能となるため、パラメータ設定を簡便化できるとともに、初心者ユーザにも簡単に使用できる。

【0016】また、アプリケーションが出力するデータの内容を解析し、その解析結果に基づいてアプリケーションからの出力データ毎にシリアルインタフェースのパラメータを変更することで、1文字を送信する際に必要なビット数が最少になるようにパラメータを設定でき

る。その結果、実通信時間を短くできる。さらに、ターゲットマシンとしてプリンタを用いたシステムにおいて、このプリンタでは、ホストコンピュータが出力した印刷データからホストコンピュータ側のシリアルインタフェースのパラメータを解析し、この解析結果に合わせてパラメータを設定することで、パラメータを一切設定することなく、任意のホストコンピュータとの間でデータの送受信が可能となる。

【0017】

【実施例】以下、本発明の実施例を図面に基づいて詳細に説明する。図1は、本発明の第1実施例を示す機能ブロック図である。図1において、ホストコンピュータ11とターゲットマシンである例えばプリンタ12とは、シリアルインタフェースであるRS232C I/F 13によって接続されて相互に送受信可能となっている。ホストコンピュータ11は、RS232C I/F 13のパラメータをプリンタ12との間で予め決められたパラメータ（以下、システム・パラメータと称する）に設定するシステム・パラメータ設定処理部14と、データを送信するためのパラメータ（以下、データ通信パラメータと称する）を設定するデータ通信パラメータ決定処理部15と、最終的にRS232C I/F 13のパラメータを設定するI/Fパラメータ設定処理部16とを有している。

【0018】このホストコンピュータ11では、AUTO/MANUALモード選択部17において、AUTOモードが選択されたときには、I/Fパラメータの設定が自動的に行われ、MANUALモードが選択されたときには、I/Fパラメータの設定が従来技術と同様にマニュアルにて行われるようになっている。プリンタ12は、RS232C I/F 13のパラメータをシステム・パラメータに設定するシステム・パラメータ設定処理部18と、受信したデータ通信パラメータが設定可能であるかどうかを判定するパラメータ設定判定処理部19と、最終的にRS232C I/F 13のパラメータを設定するI/Fパラメータ設定処理部20とを有している。このプリンタ12でも、AUTO/MANUALモード選択部21でモード選択が行われる。

【0019】次に、上記第1実施例におけるホストコンピュータ上のRS232C I/Fドライバの動作の流れを、図2および図3のフローチャートにしたがって説明する。まず、ホストコンピュータの起動と同時に、DOSによってホストコンピュータを初期化し（ステップS1）、次いでSPEEDコマンドによってRS232C I/Fドライバを起動することにより、RS232C I/F回路およびRS232C I/Fプログラムの初期化を行ってRS232C I/Fを使用可能にする（ステップS2）。この状態において、ユーザによってAUTOモード又はMANUALモードの選択が行われる。

【0020】このモード選択において、AUTOモードが選択されると（ステップS3）、現在設定されている

RS232C I/Fのパラメータ値をDOSから読み込んでメモリ上に格納し（ステップS4）、続いてRS232C I/Fのパラメータをシステム・パラメータに設定する（ステップS5）。このシステム・パラメータは、新しい設定値を送信するためのパラメータであり、一例として、次のようなパラメータを設定する。すなわち、BAUD RATE/300 bps, DATA BIT/8 bits, PARITY/NONE, STOP BIT/1 bit, FLOW CONTROL/DTR HI である。

【0021】次に、ターゲットマシンがシステム・パラメータに設定され、通信が可能か否かを判断するために、送信開始コード（例えば12h, “h”は16進数であることを示す）を、RS232C I/Fで接続されているターゲットマシンに送信する（ステップS6）。そして、ある一定時間ターゲットマシンからのACKの送信を待つ（ステップS7, S8）。ACKを受信すると、続いて受信データがターゲットマシンから送られたRS232C I/Fの設定可能なパラメータのリスト（以下、パラメータテーブルと称する）であるか否かをある一定時間監視する（ステップS9, S10）。

【0022】次に、受信したパラメータテーブルと、DOSが持っているRS232C I/Fのパラメータテーブルを比較し、最適なデータ通信パラメータを決定し（ステップS11）、この決定したデータ通信パラメータをターゲットマシンに送信する（ステップS12）。そして、ターゲットマシンからのACKもしくはNACKの受信をある一定時間待つ（ステップS13～S15）。受信したデータがACKの場合は、データ通信パラメータにしたがってRS232C I/F回路およびドライバを設定する（ステップS16）。受信したデータがNACKの場合は、ステップS9に戻って再びパラメータテーブルの受信を待つ。

【0023】ステップS8, S10, S15の受信待ちの状態において、ある一定時間が経過し、タイムアウトと判定した場合は、ステップS4で読み出したRS232C I/FのパラメータにしたがってRS232C I/F回路およびドライバを設定し（ステップS17）、続いてオペレータに対してエラー表示を行い（ステップS18）、一連の処理を終了する。なお、ステップS3でのMANUALモードの選択の場合には、MANUALモードの表示を行い（ステップS19）、しかる後マニュアルにてパラメータを設定する（ステップS20）。このパラメータの設定は、図17に示す従来技術と同様の処理（ステップS203～S207）にて行われる。

【0024】次に、第1実施例におけるプリンタ側の動作の流れを、図4および図5のフローチャートにしたがって説明する。図20に示す従来例と同様の処理（ステップS211～S215, S218）を経た後、ホストI/FとしてRS232C I/Fが選択されたか否かを判断し（ステップS31）、RS232C I/Fの選択でなければ、Parallel I/Fの選択であることから、Parallel I/Fを設定

し(ステップS32)、以降、図20のステップS215に移行して従来技術と同様の処理を実行する。

【0025】一方、RS232C I/Fの選択であれば、この状態において、ユーザによるAUTO/MANUALモードの選択が行われる。そして、その選択がAUTOモードであるかMANUALモードであるかの判断を行う(ステップS33)。ここで、MANUALモードが選択されると、マニュアルにてパラメータを設定する(ステップS34)。このパラメータの設定は、図21に示す従来技術と同様の処理(ステップS221~S225)によって行われる。AUTOモードが選択されると、RS232C I/Fをシステム・パラメータに設定し(ステップS35)、続いてホストコンピュータから送信開始コードが送られてくるのをある一定時間待つ(ステップS36、S37)。

【0026】送信開始コードを受信すると、ACKをホストコンピュータへ送信し(ステップS38)、続いてRS232C I/Fのパラメータテーブルをホストコンピュータに送信する(ステップS39)。そして、ホストコンピュータからデータ通信パラメータが送られてくるのをある一定時間待つ(ステップS40、S41)。データ通信パラメータを受信すると、送られてきたデータ通信パラメータとパラメータテーブルを比較することによってこのパラメータが設定可能であるか否かを判断する(ステップS42)。

【0027】ここで、設定不可能と判断した場合は、ホストコンピュータにNACKを送信し(ステップS43)、ステップS39に戻って再びパラメータを送信する。一方、設定可能と判断した場合は、受信したデータ通信パラメータにしたがってRS232C I/F回路を設定し(ステップS44)、一連の処理を終了する。ステップS37、S41の受信待ちの状態において、ある一定時間が経過し、タイムアウトと判定した場合は、予め決められているパラメータのデフォルト値にしたがってRS232C I/F回路を設定し(ステップS45)、続いてエラー表示を行い(ステップS46)、一連の処理を終了する。

【0028】上述したように、先ず、シリアルインタフェースのパラメータを予め決められたシステム・パラメータに設定して通信できるようにし、この通信可能な状態において、ターゲットマシンから受信したパラメータテーブルに基づいてデータ通信パラメータを決定してこれをターゲットマシンに送信し、このデータ通信パラメータに基づいてホストコンピュータとターゲットマシンとの間で自動的にパラメータを設定することにより、ユーザがそれぞれの機器のシリアルインタフェースのパラメータを設定しなくても通信が可能となる。

【0029】ところで、RS232C I/Fのパラメータの自由度が大きいため、通信速度とデータ化けに対する保証をトレードオフで設定できる。例えば、送信するデータが

テキストだけで構成されているデータであれば、1文字を送信する際に必要なビット数が最少になるようにパラメータを設定すれば実通信時間は短くなる。以下に、実通信時間を計算し、どの程度短縮されるかを示す。

#### ①通常の通信の場合

BAUD RATE/9600 BPS, DATA BIT/8 bits, STOP BIT/1 bit, PARITY/EVEN

上記パラメータで1 bit を送信するために必要な時間は、

$$\text{【数1】 } 1 \text{ sec}/9600 = 104 \mu\text{sec} \quad \cdots (1)$$

である。

【0030】1文字(1 byte)は、スタートビット+データビット+パリティビット+ストップビットで構成されるため、1文字は11ビットとなる。したがって、1文字(1 byte)を送信するために必要な時間は、

【数2】

$$104 \mu\text{sec} \times 11 \text{ ビット} = 1.144 \text{ msec} \quad \cdots (2)$$

となる。また、100k byte のデータを送信するために必要な時間は、

【数3】

$$100 \text{ k byte} \times 1.144 \text{ msec} = 114.4 \text{ sec} \quad \cdots (3)$$

となる。

【0031】②高速通信のためのパラメータを設定した場合

BAUD RATE/9600 BPS, DATA BIT/7 bits, STOP BIT/1 bit, PARITY/NONE

1 bit を送信するために必要な時間は、通常の通信の場合と同じである。1文字(1 byte)は、9ビットで構成されるため、1文字(1 byte)を送信するために必要な時間は、

【数4】

$$104 \mu\text{sec} \times 9 \text{ ビット} = 936 \mu\text{sec} \quad \cdots (4)$$

となる。また、100k byte のデータを送信するために必要な時間は、

【数5】

$$100 \text{ k byte} \times 936 \mu\text{sec} = 93.6 \text{ sec} \quad \cdots (5)$$

となる。したがって、(式(3) - 式(5))より、20.8 secだけ実通信時間を短縮できることがわかる。

【0032】ところが、上記の高速通信のためのパラメータでは、DATA BITが7ビットのため、ASCIIコード表の7F hまでの、テキストデータやポストスクリプト(PostScript)データ(印刷データが全てテキストで表記されるコマンド体系を持つ)は正常に印刷可能であるのに対して、グラフィックデータおよびソフトウェアフォントを使用するような印刷データでは、ASCIIコード表の80 h~FF hまでのデータを送信するため、DATA BITを8ビットに設定しなければ正常に印刷できない。

【0033】そこで、本発明においては、RS232C I/Fのパラメータを送信するデータ毎に変更することにより、実通信時間を短縮するようにする。図6は、本発明の第

2実施例を示す機能ブロック図である。この第2実施例においては、ホストコンピュータ11側に、ワードプロセッサ等のアプリケーションから出力されるデータの内容を解析するデータ解析処理部22が追加されている。そして、I/Fパラメータ設定処理部16は、データ解析処理部22での解析結果に基づいてアプリケーションからの出力データ毎にパラメータを変更するようになっている。

【0034】次に、上記第2実施例におけるホストコンピュータ上のRS232C I/Fドライバの動作の流れを、図7のフローチャートにしたがって説明する。本フローチャートにおいて、ステップS16でデータ通信パラメータにしたがってRS232C I/F回路およびドライバを設定するまでの処理は、第1実施例の場合と同様である。データ通信パラメータの設定後、本ドライバはメモリ上に常駐し、印刷データのRS232C I/Fへの出力か、常駐解除コマンドの発行を待つ(ステップS51、S52)。本ドライバの常駐を終了する常駐解除コマンドが入力された場合は、一連の処理を終了する。

【0035】ワードプロセッサ等のアプリケーションから印刷データがRS232C I/Fに出力された場合は、現在DOSに設定されているRS232C I/Fのパラメータ(設定値)を読み出し(ステップS53)、続いてその印刷データに80h~Ffhのデータが存在するかどうかを調べる(ステップS54)。そして、ステップS53で読み出した現在のRS232C I/Fのパラメータ値と、ステップS54で調査した結果から、RS232C I/Fのパラメータ値を変更するかどうかを判断し(ステップS55)、変更する必要がある場合は、印刷データを送信し(ステップS56)、しかる後ステップS51に戻って再び印刷データの出力と常駐解除コマンドの発行を待つ。

【0036】一方、変更する必要がある場合は、送信開始コードをターゲットマシンへ送信し(ステップS57)、ACKの受信をある一定時間待つ(ステップS58、S59)。ACKを受信すると、新たなデータ通信パラメータをターゲットマシンへ送信し(ステップS60)、ACKの受信をある一定時間待つ(ステップS61、S62)。ACKを受信すると、印刷データをターゲットマシンへ送信し(ステップS63)、しかる後ステップS51に戻って再び印刷データの出力と常駐解除コマンドの発行を待つ。ステップS59、S62の受信待ちの状態において、ある一定時間が経過し、タイムアウトと判定した場合は、そのままステップS63に移行する。

【0037】次に、第2実施例におけるプリンタ側の動作の流れを、図8のフローチャートにしたがって説明する。まず、現在選択されているI/Fに受信データがあるかどうかを判断し(ステップS71)、受信データがない場合は、図20に示す従来技術と同様の処理(ステップS213~S217)を行う。受信データがある場合

は、受信したデータが送信開始コードであるか否かを判断する(ステップS72)。そして、送信開始コードでない場合は、従来技術と同様の印刷処理を行う(ステップS73)。

【0038】一方、送信開始コードであると判定した場合は、ACKをホストコンピュータに送信し(ステップS74)、続いてデータ通信パラメータの受信をある一定時間待つ(ステップS75、S76)。この受信待ち状態において、ある一定時間が経過し、タイムアウトと判定した場合は、エラーの表示を行い(ステップS77)、一連の処理を終了する。データ通信パラメータを受信すると、メモリ上に格納されているパラメータテーブルと比較し(ステップS78)、設定可能であるか否かを判断する(ステップS79)。

【0039】ここで、設定不可能と判定した場合は、ホストコンピュータへNACKを送信し(ステップS80)、しかる後ステップS75に戻って再びデータ通信パラメータの受信待ちとなる。一方、設定可能であると判断した場合は、ホストコンピュータへACKを送信し(ステップS81)、新たなデータ通信パラメータをRS232C I/F回路に設定し(ステップS82)、一連の処理を終了し、再びステップS71に戻ってデータの受信待ちとなる。

【0040】上述したように、ワードプログラムなどのアプリケーションが出力するデータの内容を解析し、その解析結果に基づいてアプリケーションからの出力データ毎にシリアルインタフェースのパラメータを変更することにより、1文字を送信する際に必要なビット数が最少になるように当該パラメータを設定できるので、実通信時間を短くできる。

【0041】図9は、本発明の第3実施例を示す機能ブロック図である。この第3実施例は、ターゲットマシンとしてプリンタを用いたシステムへの適用に限定されるものである。この第3実施例において、プリンタ12は、ホストコンピュータ11が出力した印刷データからホストコンピュータ11側のRS232C I/Fのパラメータを解析するパラメータ解析処理部22を有し、I/Fパラメータ設定処理部20にてパラメータ解析処理部22の解析結果に合わせてパラメータを設定するようになっている。

【0042】次に、上記第3実施例におけるプリンタ側の動作の流れを、図12~図16のフローチャートにしたがって説明する。図20に示す従来例と同様の処理(ステップS211~S215、S218)を経た後、ホストI/FとしてRS232C I/Fが選択されたか否かを判断し(ステップS101)、RS232C I/Fの選択でなければ、Parallel I/Fの選択であることから、Parallel I/Fを設定し(ステップS102)、以降、図21のステップS215に移行して従来技術と同様の処理を実行する。



【0043】一方、RS232C I/Fの選択であれば、続いて ANALYZEモードを選択したか否かを判断する(ステップS103)。ここで、ANALYZEモードを選択すると、RS232C I/Fパラメータの1つであるボーレート(Baud Rate)を例えば38400bpsに設定する(ステップS104)。そして、ON LINE状態となり、表示部に ON LINE表示を行う(ステップS105)。続いて、受信データ線の信号が変化したか、もしくは ON-LINEキーが押されたかを無限ループにて監視する(ステップS106、S107)。そして、ON-LINEキーが押されたと判断すると、OFF LINE状態となり、OFF LINE表示を行う(ステップS108)。

【0044】一方、受信データ線の信号が変化したと判定すると、メモリ上に設定された解析フラグがセットされているか否かを判断し(ステップS109)、解析フラグがセットされていると判定すると、従来技術と同様な印刷処理を行う(ステップS110)。また、解析フラグがセットされていないと判定すると、ホストコンピュータから送られるデータを上記ボーレートでサンプリングし、データの変化点(“0”→“1”, “1”→“0”)から変化点までの時間を2Byteずつメモリ上に設定された領域に書き込み(ステップS111)、このサンプリングを10秒間行う(ステップS112)。10秒間データを記録した後、記録を続けながら、フロー・コントロールの解析を行う(図10参照)。

【0045】ここで、一例として、ホストコンピュータから以下のパラメータで送信される場合を考える。すなわち、BAUD RATE/19200 bps, FLOW CONTROL/DTR HI, DATA BIT/7 bits, PARITY/EVEN, STOP BIT/2 bitsである。そして、ホストコンピュータから送られるデータの例を、「Test Sample 1, ^L」とすると、実際に送信されるデータの内容は、“54 65 73 74 20 53 61 6D 70 65 20 31 2E 0C 0D 0A”となり、サンプリングした結果のメモリ上のデータは、“0082 0034 0068 0034 0034 0034 0034 0068 0138 0068 0034 0034 ……”となる。

【0046】続いて、RS232C I/Fで規定されているDTR信号を反転して出力し(ステップS113)、受信データ線の信号の変化でホストコンピュータからのデータ送信が止まったか否かを判断する(ステップS114)。ここで、データ送信が止まらなと判定すると、DRT信号を元の状態に戻し(ステップS115)、RS232C I/Fで規定されているSSD信号を反転して出力する(ステップS116)。次いで、再びホストコンピュータからのデータ送信が止まったか否かを判断し(ステップS117)、止まらなと判定すると、SSD信号を元の状態に戻し(ステップS118)、Xoff(13h)をホストコンピュータへ送信する(ステップS119)。

【0047】次に、ホストコンピュータからのデータ送信が止まったか否かを判断し(ステップS120)、データ送信が止まると判定すると、ステップS113～

S119で行った調査結果から、プリンタのフロー・コントロール(FLOW CONTROL)を設定する(ステップS121)。そして、ステップS111で記録したデータの中から最も短いパルス幅のデータを調べ(ステップS112)、例えば図11に示すような、設定可能なボーレートとそのボーレートのパルス幅の対応表(パルス幅テーブル)を参照して最も近いボーレートを設定する(ステップS123)。

【0048】次に、ステップS111で記録したデータをステップS122で調べた最も短いパルス幅(本例では、0034)で割ることにより、“1”と“0”のデータ“0010010101100000001101”に変換し、メモリ上に格納する(ステップS124)。本例では、9ビット目から“00……01”のパターンが始まっている。続いて、ステップS124で格納したデータから、最初の“1”(スタート・ビットを示す)を除いて、8ビット目から10ビット目の間で、スタート・ビット、ストップ・ビットのパターン(“00……01”のパターンであり、以下、SSパターンと称する)が最初に始まるビット位置を記録し(ステップS125)、SSパターンの次のビットから数えて同じ位置にSSパターンがあるか否かを判断する(ステップS126)。

【0049】ここで、SSパターンがないと判定すると、最初のデータに戻り、ステップS125で調べた位置の次のビットから調べてSSパターンを探し、そのビット位置を記録する(ステップS127)。SSパターンがあると判定すると、続いて最終データまで調べたか否かを判断し(ステップS128)、最終データまで調べていなければ、ステップS126に戻って同じ処理を繰り返す。最終データまで調べていれば、続いてSSパターンが始まるビットは8ビット目であるか否かを判断し(ステップS129)、8ビット目から始まると判定すると、Data Bit=7 Bit、Parity=NONEに設定する(ステップS130)。

【0050】また、ステップS131において、SSパターンが9ビット目から始まると判断すると、メモリ上に設定された偶数フラグと奇数フラグをクリアし(ステップS132)、次にデータの読出しアドレスを先頭に戻し(ステップS133)、その先頭からスタート・ビットを除いて7ビット単位でデータを読み込み(ステップS134)、しかる後、読み込んだデータに対して8ビット目が偶数パリティであるか否かを判断する(ステップS135)。そして、偶数パリティであれば偶数フラグをセットし(ステップS136)、偶数パリティでなければ奇数フラグをセットする(ステップS137)。

【0051】次に、ステップS124で書き込んだデータを最終データまで読み込んだか否かを判断し(ステップS138)、最終データまで読み込んでいないと判定

すると、次のデータを7ビット読み込み(ステップS139)、しかる後ステップS135に戻って再び8ビット目のパリティを調べる。一方、最終データまで読み込んだと判定すると、偶数フラグがセットされているか否かを判断する(ステップS140)。そして、偶数フラグがセットされていなければ、Data Bit=7Bit、Parity=ODDに設定し(ステップS141)、偶数フラグがセットされていれば、奇数フラグがセットされているか否かを判断する(ステップS142)。ここで、奇数フラグがセットされていないと判定すると、Data Bit=7Bit、Parity=ENENに設定に設定する(ステップS143)。ステップS142でも、奇数フラグがセットされていると判定すると、Data Bit=8Bit、Parity=NONEに設定に設定する(ステップS144)。

【0052】ステップS131において、SSパターンが10ビット目から始まると判定すると、データ・ビットを8ビットに設定し(ステップS145)、続いて8ビットでデータを読み込み、9ビット目が偶数パリティであるか否かを判断する(ステップS146)。ここで、偶数パリティでない判定すると、Parity=ODDに設定し(ステップS147)、偶数パリティであると判定すると、Parity=EVENに設定する(ステップS148)。

【0053】パラメータの解析が終了すると、解析フラグをセットし(ステップS149)、ステップS124で書き込んだデータを、設定されたパラメータ値にしたがって1Byteのデータに変換し、1Byteずつメモリ上に設定された受信バッファに格納する(ステップS150)。本例の場合、1文字目の内容が“1010100” (=54h)であるから、受信バッファの内容は、“54 65 73 74 20 53 61 6D 70 65 20 31 2E 0C 0D 0A”となる。

【0054】次に、設定されたフロー・コントロールより、ビジー状態を解除し、受信可能であることをホストコンピュータへ通知し(ステップS151)、続いてホストコンピュータから送られてきたデータを、設定されたパラメータにしたがって受信し、そのデータを受信バッファに格納する(ステップS152)。パラメータ値を解析できないデータを受信し場合は、RS232C I/Fのパラメータが規格外であることを示す“FLAMING ERROR”を表示し(ステップS153)、予め決められたパラメータ値(デフォルト値)にしたがってRS232C I/Fパラメータを設定し(ステップS154)、続いて解析フラグをクリアし(ステップS155)、しかる後ステップS106に戻ってデータ受信を待つ。

【0055】

【発明の効果】以上詳細に説明したように、本発明によれば、シリアルインタフェースのパラメータを予め決められたシステム・パラメータに設定して通信できるようにし、この通信可能な状態において、ターゲットマシン

から受信したパラメータテーブルに基づいてデータ通信パラメータを決定してこれをターゲットマシンに送信し、このデータ通信パラメータに基づいてホストコンピュータとターゲットマシンとの間で自動的にシリアルインタフェースのパラメータを設定するようにしたことにより、ユーザがそれぞれの機器のシリアルインタフェースのパラメータを設定しなくても通信が可能となるため、パラメータ設定を簡便化できるとともに、初心者ユーザにも簡単に使用できることになる。

【0056】また、アプリケーションが出力するデータの内容を解析し、その解析結果に基づいてアプリケーションからの出力データ毎にシリアルインタフェースのパラメータを変更するようにしたことにより、1文字を送信する際に必要なビット数が最少になるようにパラメータを設定できるので、実通信時間を短くできることにもなる。さらに、ターゲットマシンとしてプリンタを用いたシステムにおいて、このプリンタでは、ホストコンピュータが出力した印刷データからホストコンピュータ側のシリアルインタフェースのパラメータを解析し、この解析結果に合わせてパラメータを設定することにより、パラメータを一切設定することなく、任意のホストコンピュータとの間でデータの送受信が可能となる。

【図面の簡単な説明】

【図1】本発明の第1実施例を示す機能ブロック図である。

【図2】第1実施例におけるホストコンピュータ上のRS232C I/Fドライバの動作の流れを示すフローチャート(その1)である。

【図3】第1実施例におけるホストコンピュータ上のRS232C I/Fドライバの動作の流れを示すフローチャート(その2)である。

【図4】第1実施例におけるプリンタ側の動作の流れを示すフローチャート(その1)である。

【図5】第1実施例におけるプリンタ側の動作の流れを示すフローチャート(その2)である。

【図6】本発明の第2実施例を示す機能ブロック図である。

【図7】第2実施例におけるホストコンピュータ上のRS232C I/Fドライバの動作の流れを示すフローチャートである。

【図8】第2実施例におけるプリンタ側の動作の流れを示すフローチャートである。

【図9】本発明の第3実施例を示す機能ブロック図である。

【図10】第3実施例の動作説明のためのタイミングチャートである。

【図11】ボーレートとそのパルス幅の対応関係を示すパルス幅テーブルである。

【図12】第3実施例におけるプリンタ側の動作の流れを示すフローチャート(その1)である。

15

16

【図13】第3実施例におけるプリンタ側の動作の流れを示すフローチャート（その2）である。

【図14】第3実施例におけるプリンタ側の動作の流れを示すフローチャート（その3）である。

【図15】第3実施例におけるプリンタ側の動作の流れを示すフローチャート（その4）である。

【図16】第3実施例におけるプリンタ側の動作の流れを示すフローチャート（その5）である。

【図17】従来例におけるホストコンピュータ上の動作の流れを示すフローチャートである。

【図18】プリンタにおける制御部のブロック図である。

【図19】プリンタにおける操作部の正面図である。

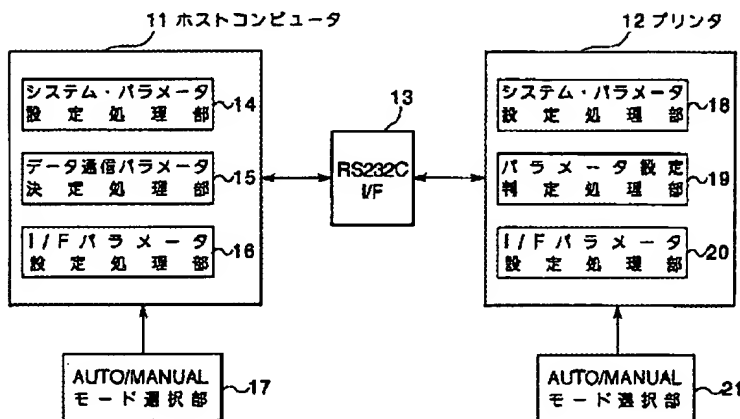
【図20】従来例におけるプリンタ側の動作の流れを示すフローチャート（その1）である。

【図21】従来例におけるプリンタ側の動作の流れを示すフローチャート（その2）である。

【符号の説明】

- 11 ホストコンピュータ
- 12 プリンタ
- 13 RS232C I/F
- 14 システム・パラメータ設定処理部
- 15 データ通信パラメータ決定処理部
- 16, 18 I/Fパラメータ設定処理部
- 22 データ解析処理部
- 23 パラメータ解析処理部

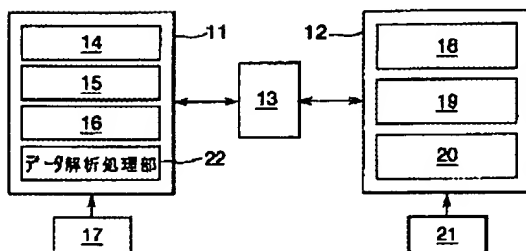
【図1】



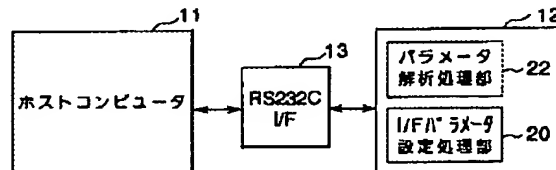
第1実施例を示す機能ブロック図

【図6】

【図9】

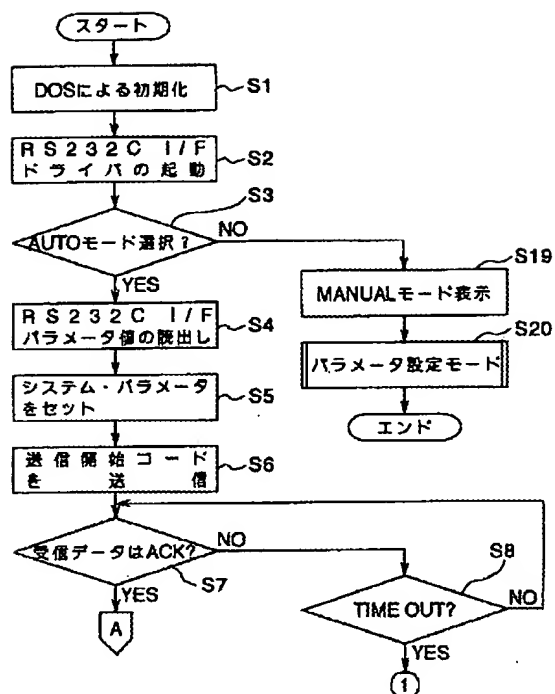


第2実施例を示す機能ブロック図



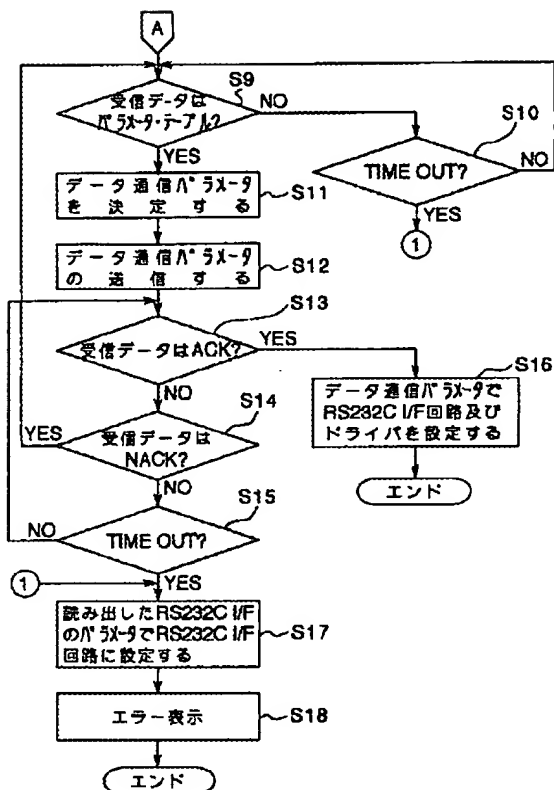
第3実施例を示す機能ブロック図

【図2】



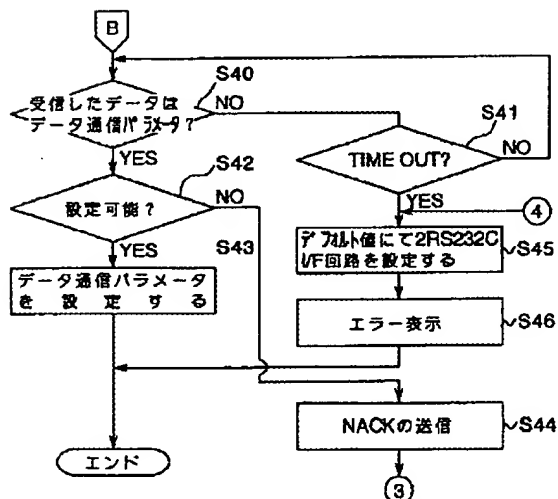
第1実施例のホストコンピュータ上の流れ図(その1)

【図3】



第1実施例のホストコンピュータ上の流れ図(その2)

【図5】



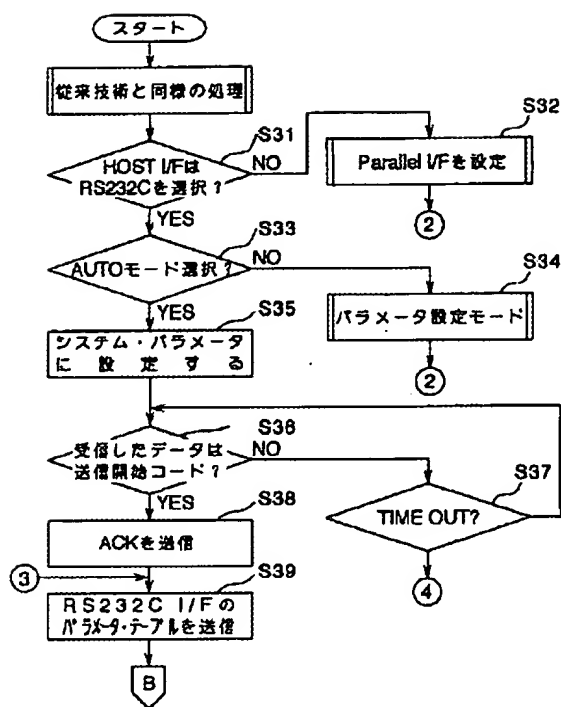
第1実施例のプリンタ上の流れ図(その2)

【図11】

| Baud Rate (bps) | Pulse (Hex) (μSEC) | Pulse (Decimal) (μSEC) |
|-----------------|--------------------|------------------------|
| 19200           | 0034               | 52                     |
| 9600            | 0068               | 104                    |
| 4800            | 00D0               | 208                    |
| 2400            | 01A0               | 416                    |
| 1200            | 0340               | 832                    |
| 600             | 0680               | 1664                   |
| 300             | 0D00               | 3328                   |

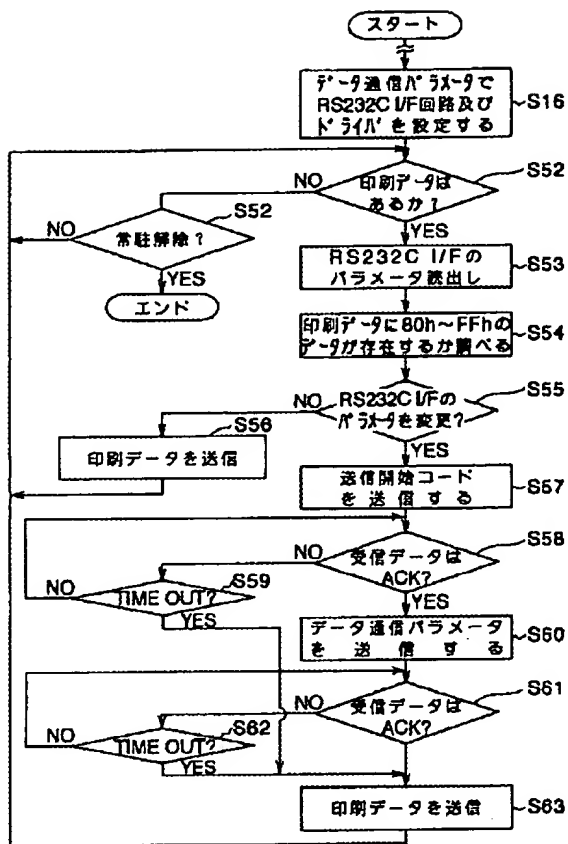
パルス幅テーブル

【図4】



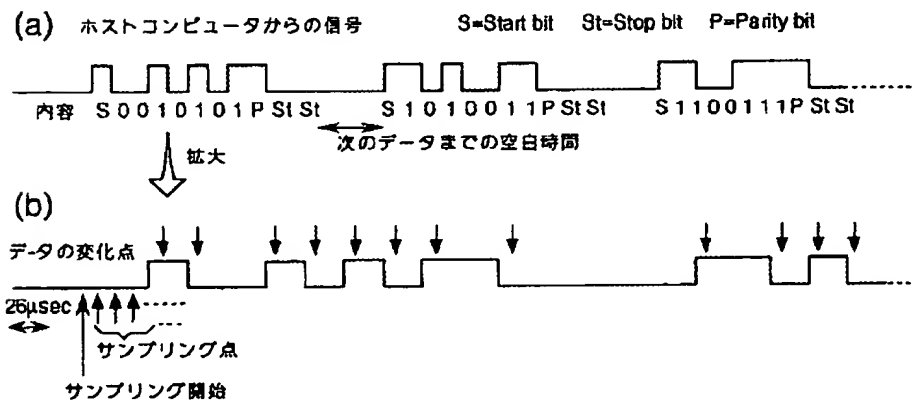
第1実施例のプリンタ上の流れ図(その1)

【図7】



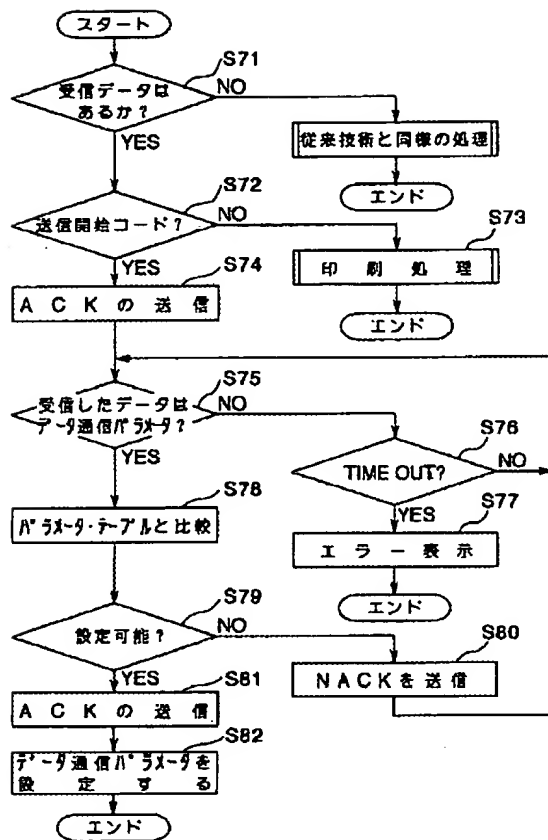
第2実施例のホストコンピュータ上の流れ図

【図10】



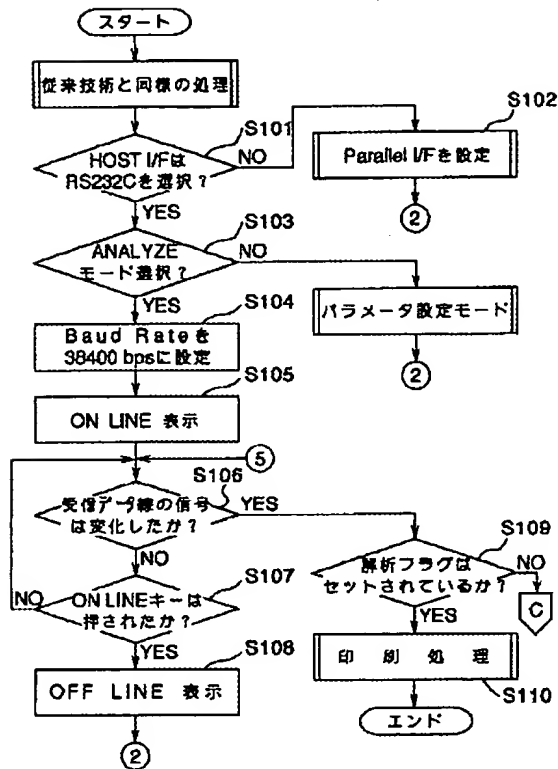
動作説明のためのタイミングチャート

【図8】



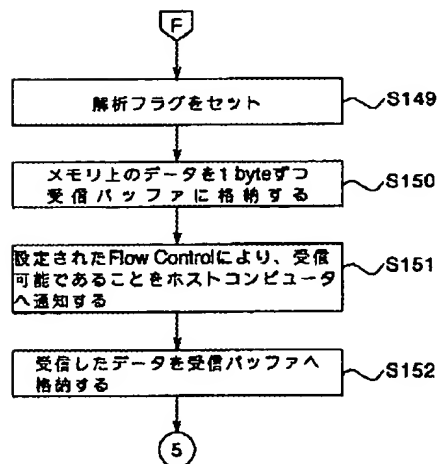
第2実施例のプリンタ上の流れ図

【図12】



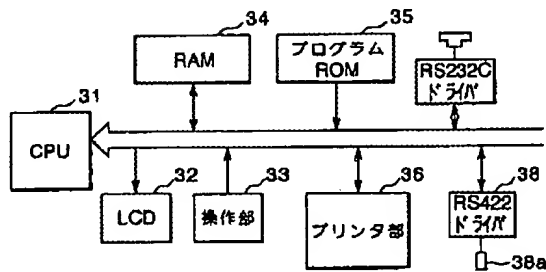
第3実施例のプリンタ上の流れ図(その1)

【図16】



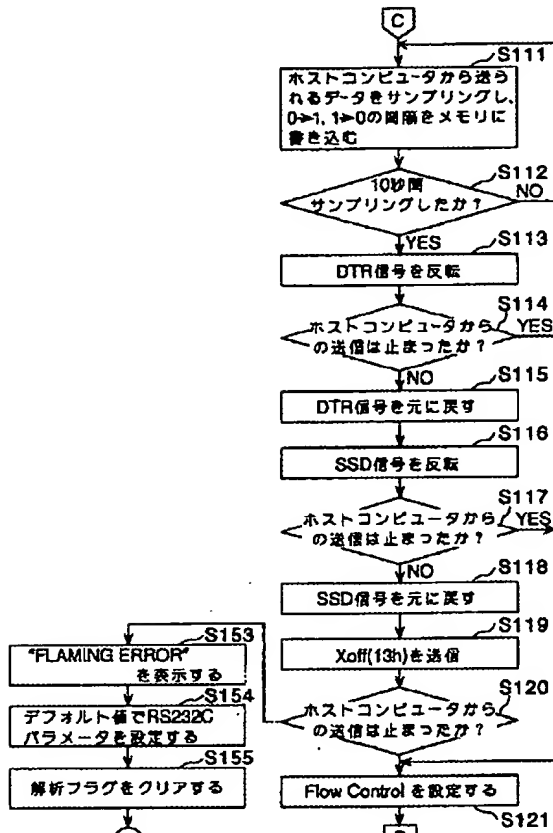
第3実施例のプリンタ上の流れ図(その5)

【図18】



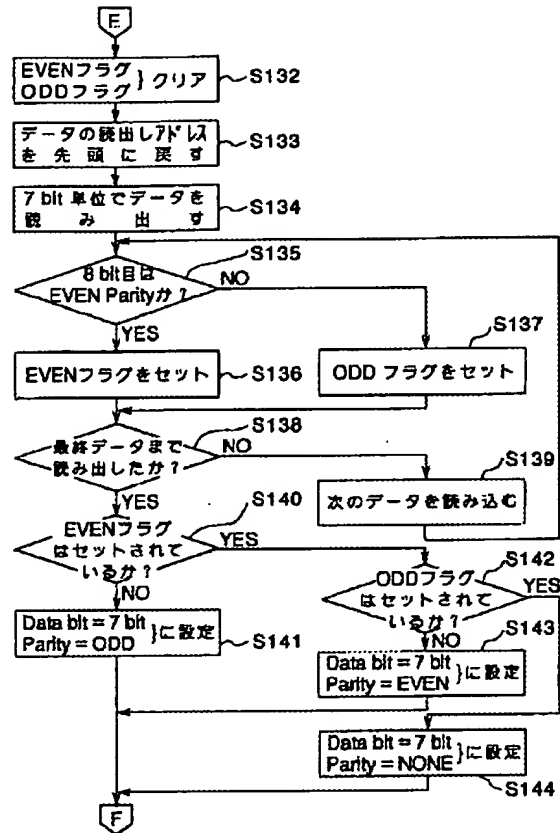
プリンタにおける制御部のブロック図

【図13】



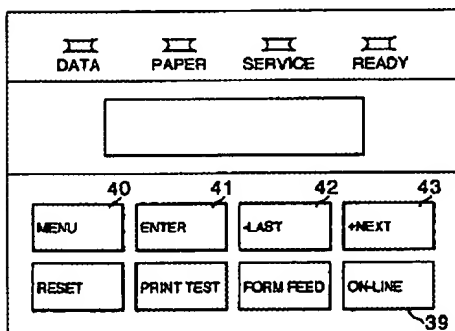
第3実施例のプリンタ上の流れ図(その2)

【図15】



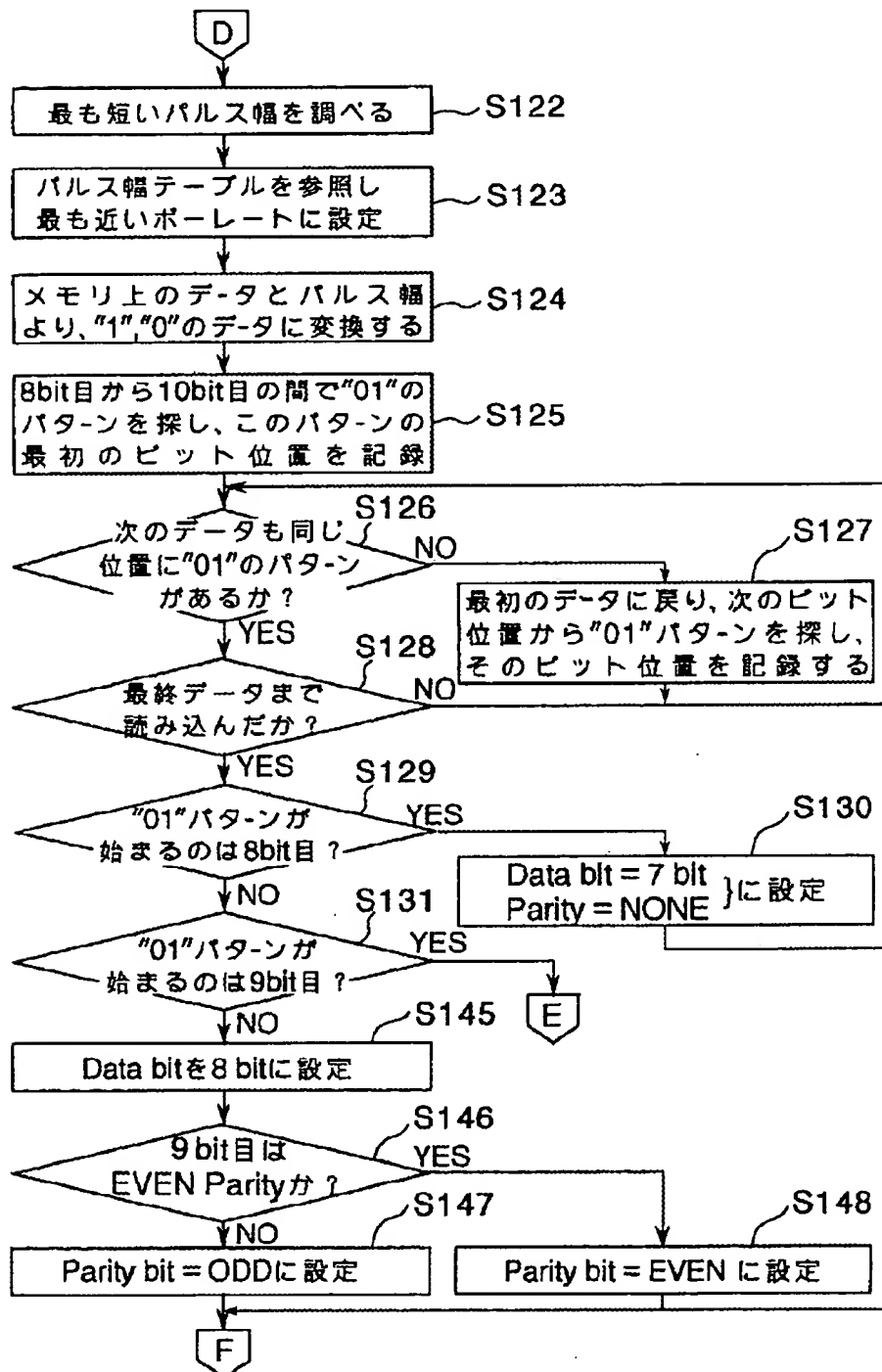
第3実施例のプリンタ上の流れ図(その4)

【図19】



プリンタにおける操作部の正面図

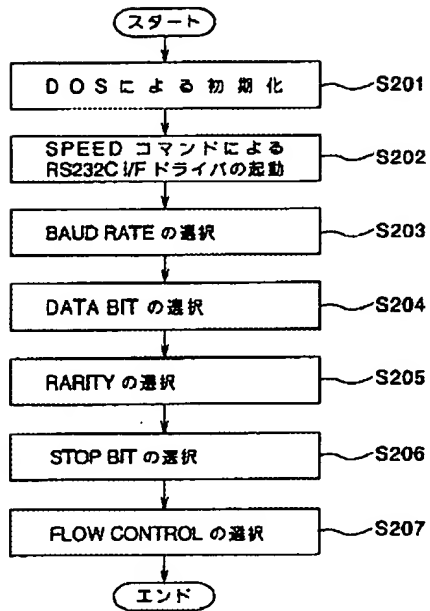
【図14】



第3実施例のプリンタ上の流れ図(その3)

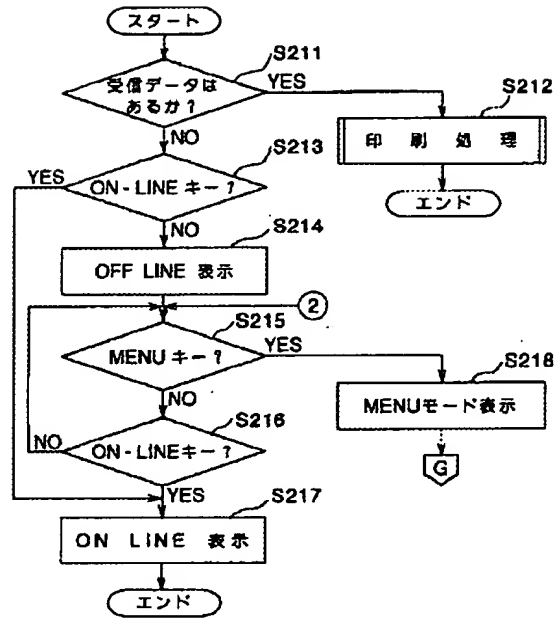


【図17】



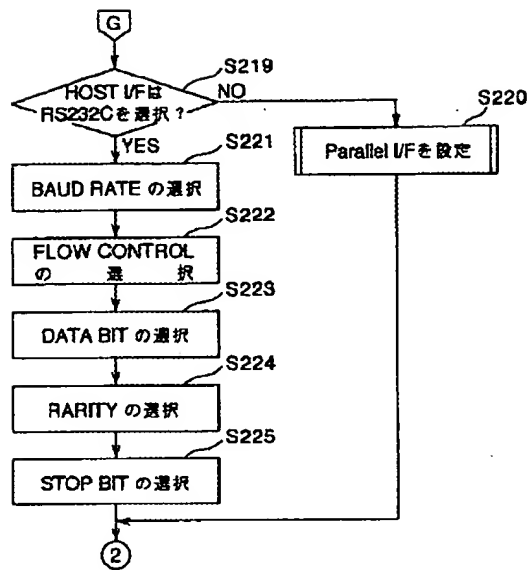
従来例のホストコンピュータ上の流れ図

【図20】



従来例のプリンタ上の流れ図(その1)

【図21】



従来例のプリンタ上の流れ図(その2)